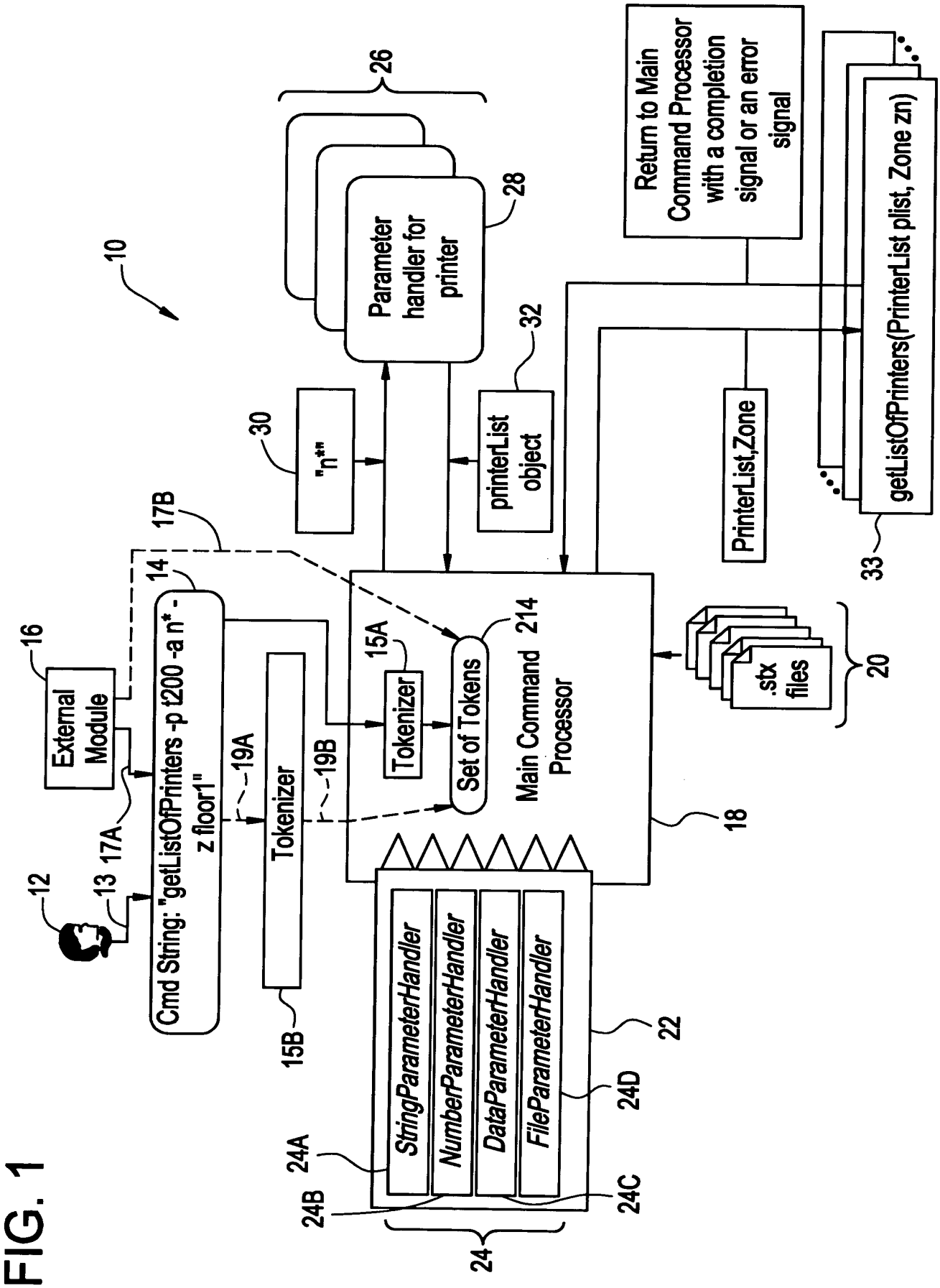

10
12
13
16
17A
17B
14
External Module
Cmd String: "getListOfPrinters -p t200 -a n* -z floor1"
19A
19B
Tokenizer
15A
Tokenizer
15B
Set of Tokens
214
Main Command Processor
18
.stx files
20
26
Parameter handler for printer
28
30
"n"
32
printerList object
PrinterList,Zone
Return to Main Command Processor with a completion signal or an error signal
getListOfPrinters(PrinterList plist, Zone zn)
33
StringParameterHandler
NumberParameterHandler
DataParameterHandler
FileParameterHandler
22
24
24A
24B
24C
24D

FIG. 1

FIG. 2

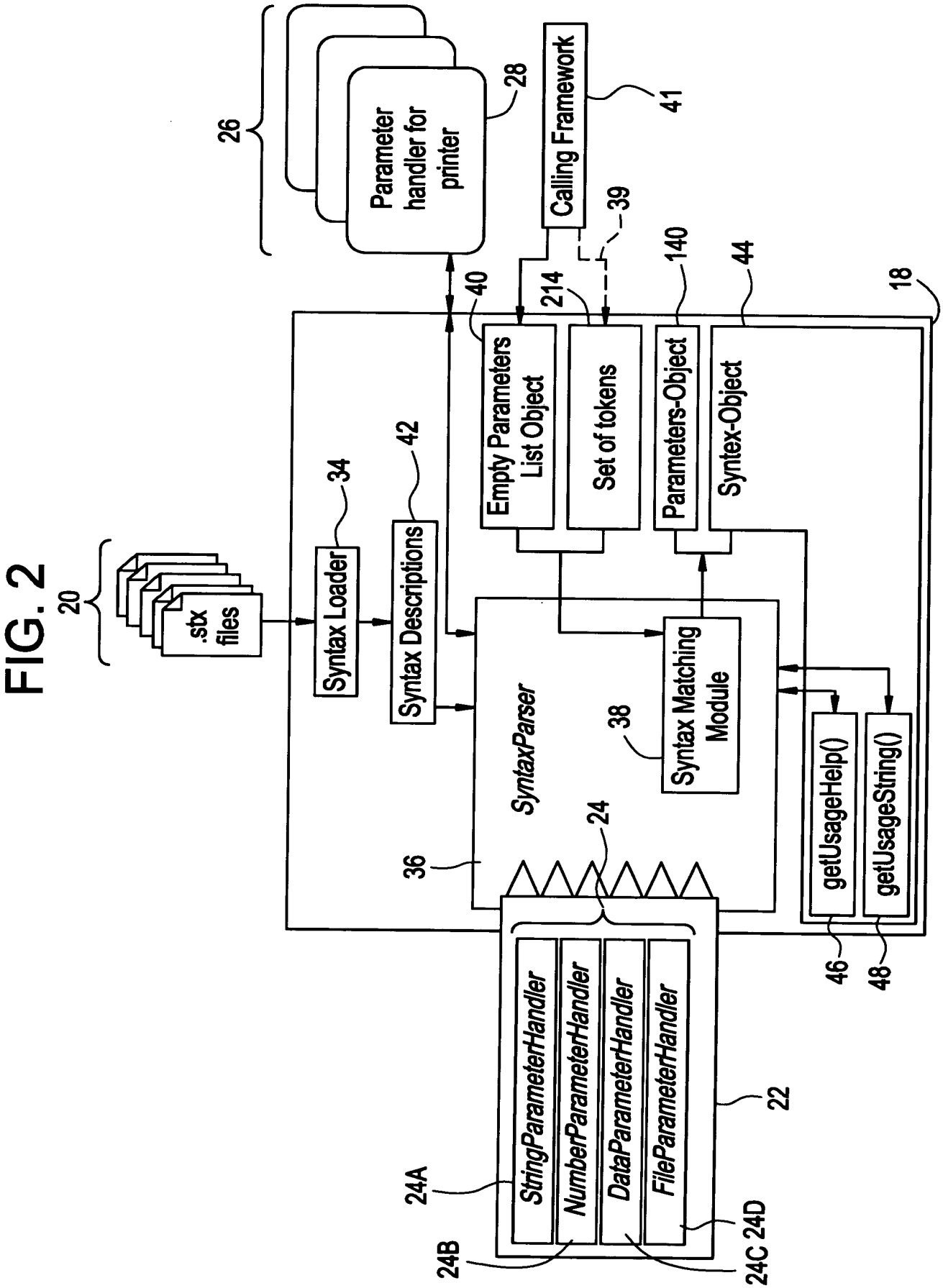

26
Parameter handler for printer
28
Calling Framework
41
39
140
44
18
40
214
Empty Parameters List Object
Set of tokens
Parameters-Object
Syntex-Object
20
.stx files
34
Syntax Loader
42
Syntax Descriptions
SyntaxParser
38
Syntax Matching Module
36
24
getUsageHelp()
getUsageString()
46
48
StringParameterHandler
NumberParameterHandler
DataParameterHandler
FileParameterHandler
22
24A
24B
24C 24D

FIG. 3

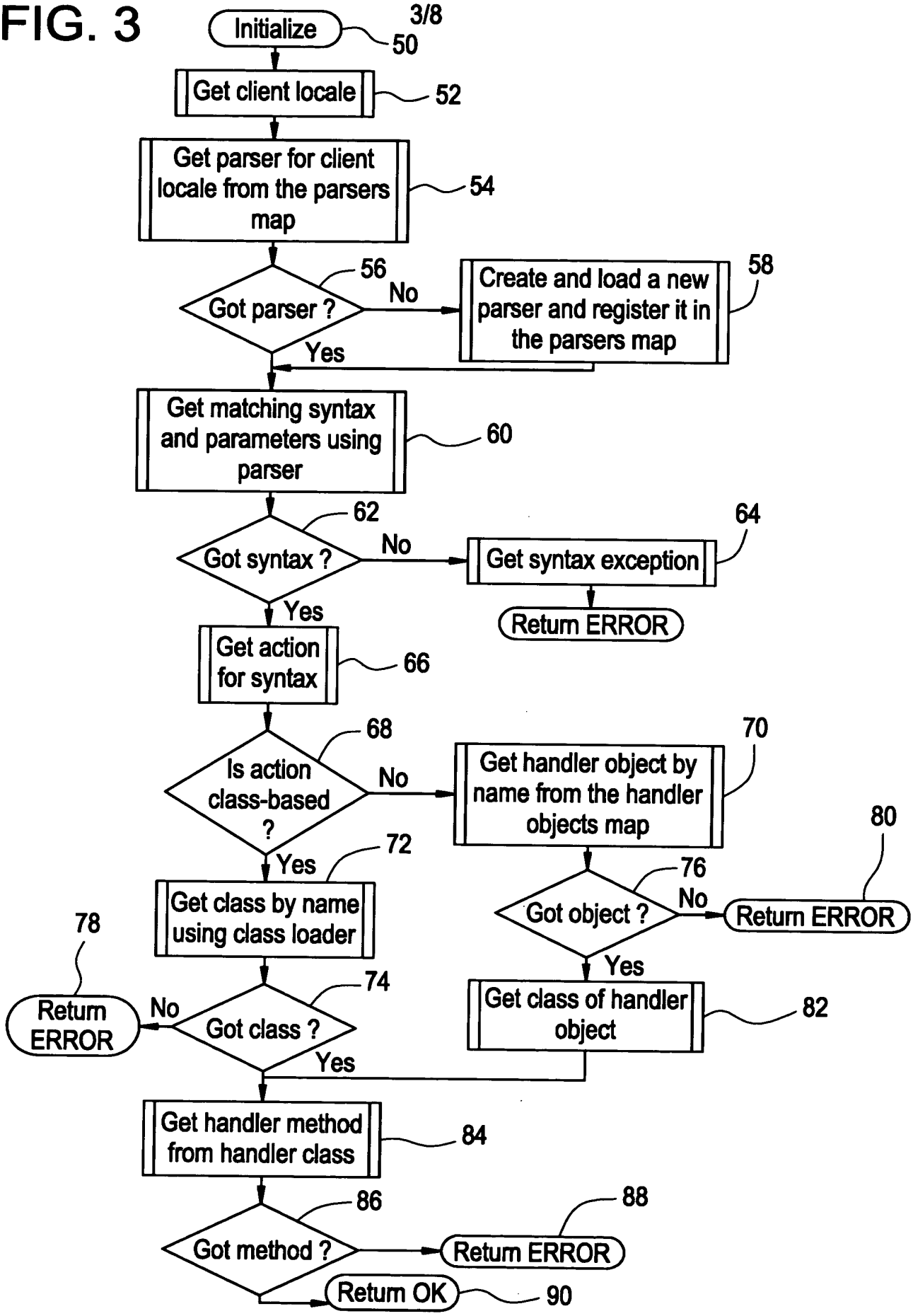

3/8
Initialize
50
Get client locale
52
Get parser for client locale from the parsers map
54
Got parser ?
56
No
Create and load a new parser and register it in the parsers map
58
Yes
Get matching syntax and parameters using parser
60
Got syntax ?
62
No
Get syntax exception
64
Return ERROR
Yes
Get action for syntax
66
Is action class-based ?
68
No
Get handler object by name from the handler objects map
70
Yes
72
Get class by name using class loader
Got object ?
76
No
Return ERROR
80
78
Return ERROR
No
Got class ?
74
Yes
Get class of handler object
82
Get handler method from handler class
84
Got method ?
86
Return ERROR
88
Return OK
90

FIG. 4

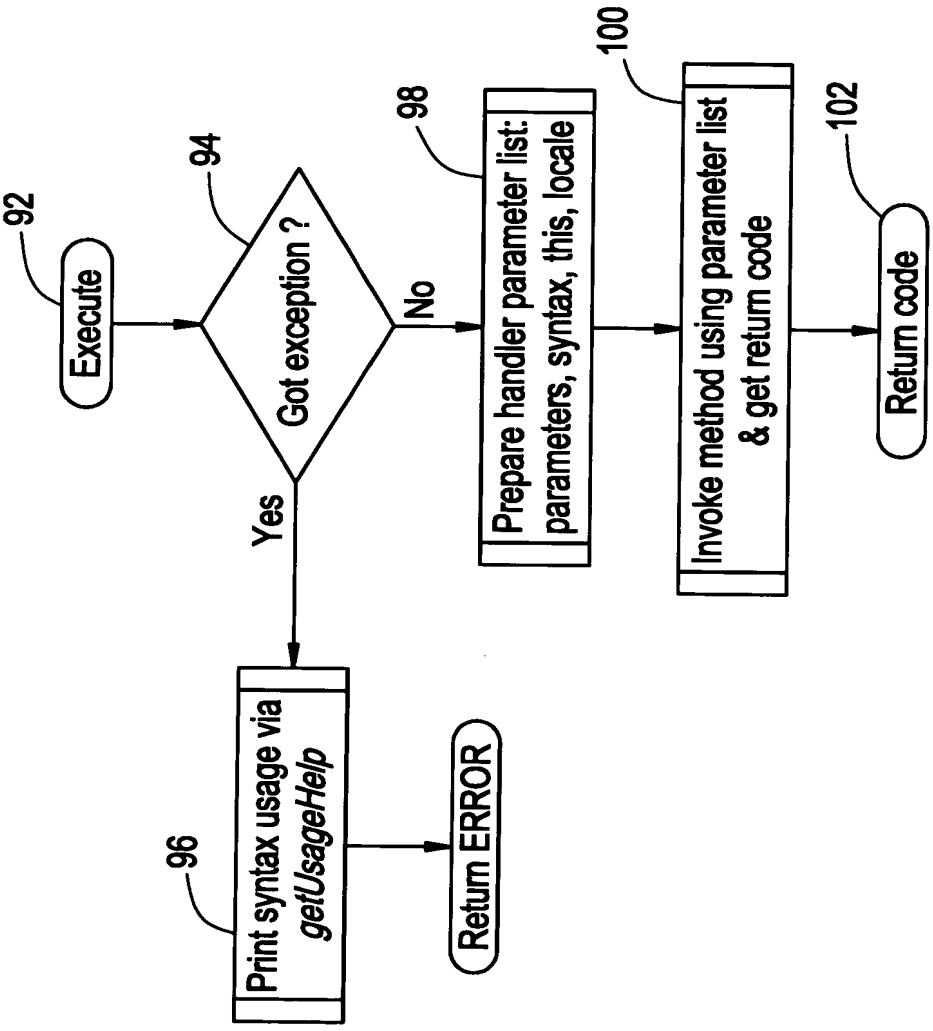

92
Execute
94
Got exception ?
No
98
Prepare handler parameter list: parameters, syntax, this, locale
100
Invoke method using parameter list & get return code
102
Return code
Yes
96
Print syntax usage via getUsageHelp
Return ERROR

FIG. 5A

104

```
[file basedefs.stx]
base
{
...
}
[file coreutils.stx]
core
{
...
}
```

FIG. 5B

106

```
Syntax Leaf Node
// Can be keywords or parameters

//parameter
zonename {
   parserType=StringParser
   description=#zoneDescription
}


//keyword
getListOfPrinters {

}
```

FIG. 5C

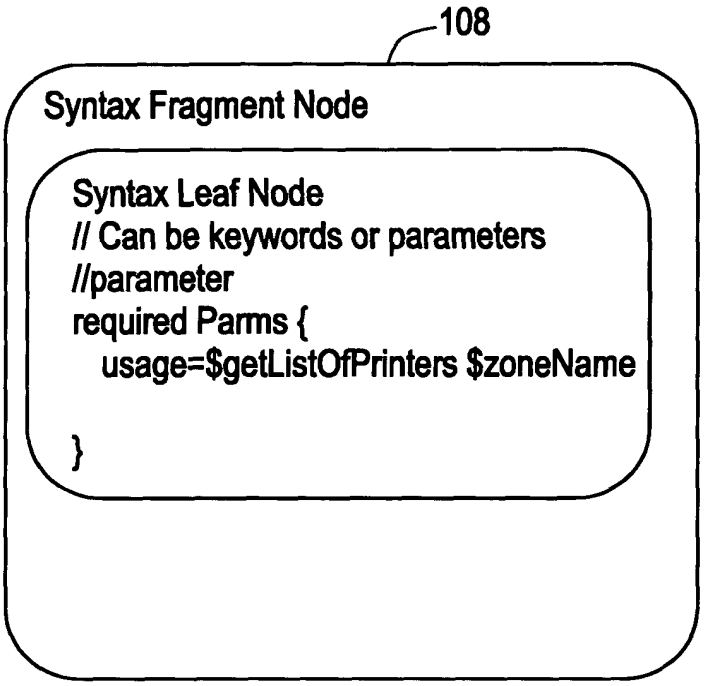

108
Syntax Fragment Node
Syntax Leaf Node
// Can be keywords or parameters
//parameter
required Parms {
usage=$getListOfPrinters $zoneName
}

FIG. 5D

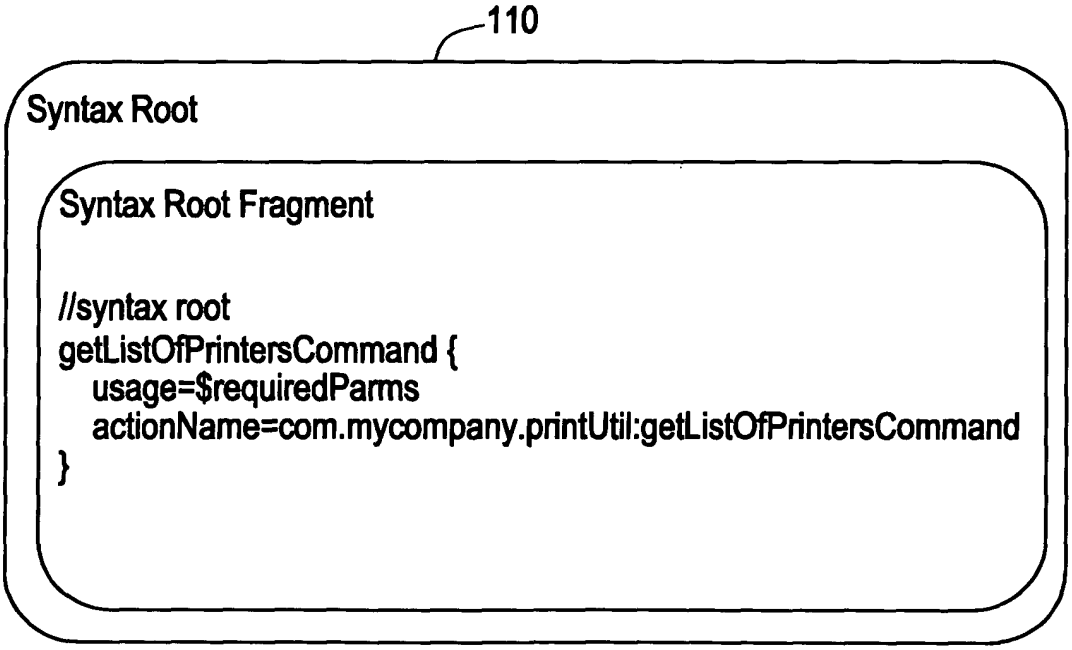
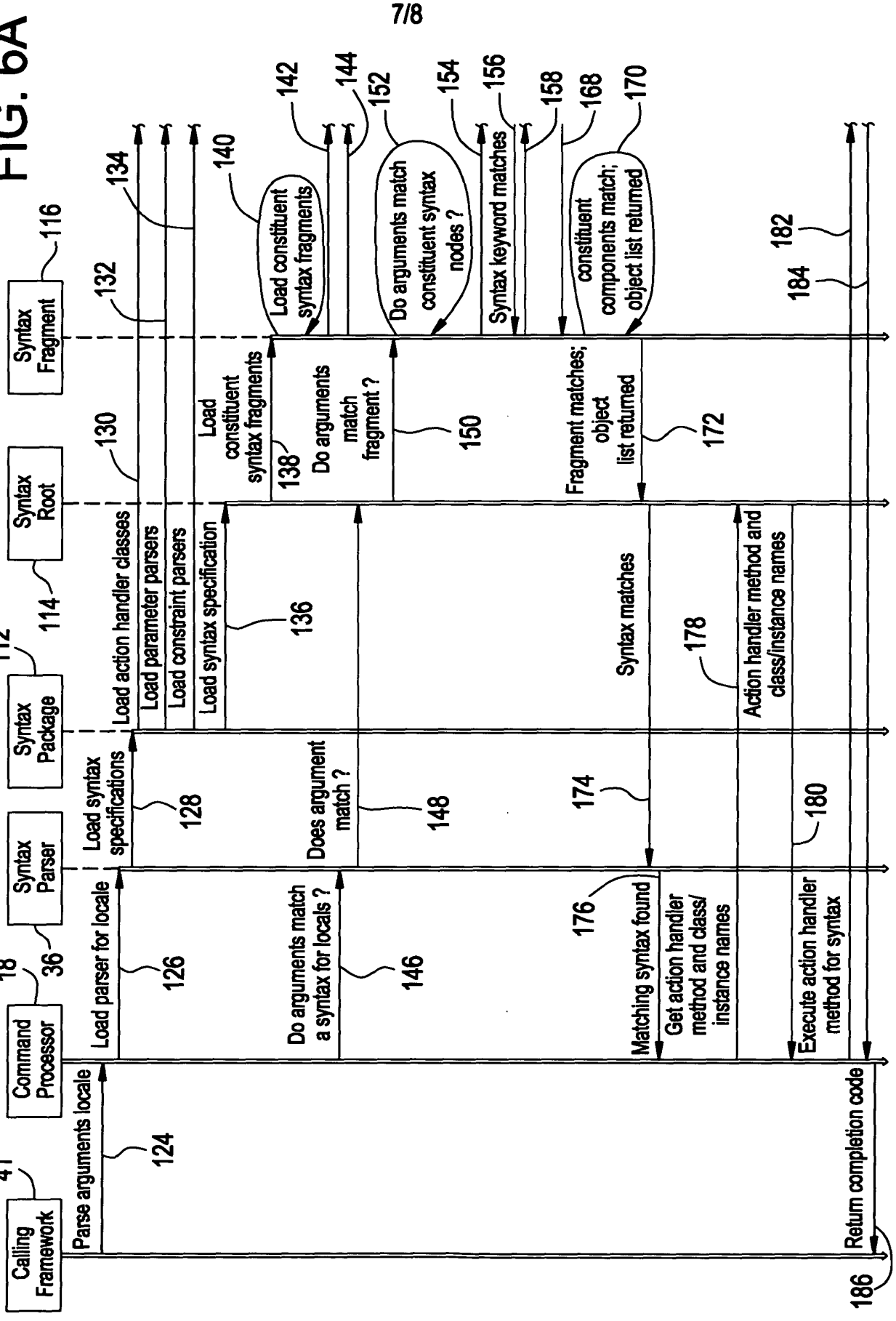

110
Syntax Root
Syntax Root Fragment
//syntax root
getListOfPrintersCommand {
usage=$requiredParms
actionName=com.mycompany.printUtil:getListOfPrintersCommand
}

FIG. 6A


Calling Framework
41
Command Processor
18
36
Syntax Parser
Syntax Package
112
Syntax Root
114
Syntax Fragment
116
Parse arguments locale
124
Load parser for locale
126
Load syntax specifications
128
Load action handler classes
130
Load parameter parsers
Load constraint parsers
Load syntax specification
132
134
136
Load constituent syntax fragments
140
138
Do arguments match a syntax for locals ?
146
Does argument match ?
148
Do arguments match fragment ?
150
Load constituent syntax fragments
142
144
Do arguments match constituent syntax nodes ?
152
154
Syntax keyword matches
156
158
Syntax matches
174
Matching syntax found
176
Get action handler method and class/ instance names
178
Fragment matches; object list returned
168
172
constituent components match; object list returned
170
Action handler method and class/instance names
180
Execute action handler method for syntax
182
184
Return completion code
186

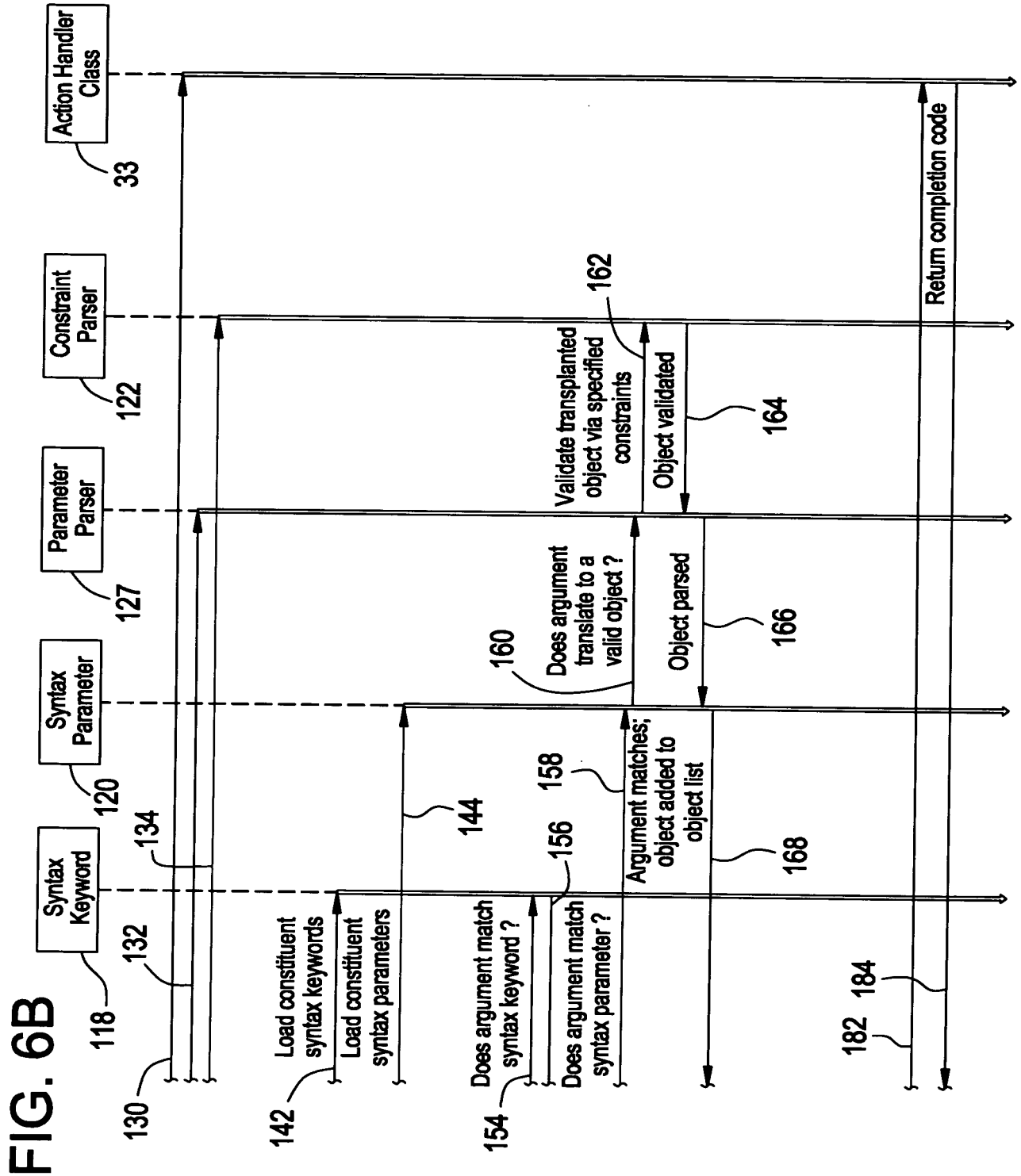

Action Handler Class
33
Constraint Parser
122
Parameter Parser
127
Syntax Parameter
120
Syntax Keyword
118
130
132
134
142
Load constituent syntax keywords
Load constituent syntax parameters
144
154
Does argument match syntax keyword ?
156
Does argument match syntax parameter ?
158
Argument matches; object added to object list
160
Does argument translate to a valid object ?
162
Validate transplanted object via specified constraints
164
Object validated
166
Object parsed
168
182
184
Return completion code

FIG. 6B